passed as an argument, and in the second case, the value of cell A4 is passed to the function. In both these cases, `IsArray` returns the value `False`.

4) If a range is passed to the function, it is passed as a two-dimensional array of values; for example, `=PositiveSum(A2:B5)`. The functions `LBound` and `UBound` are used to determine the array bounds that are used. Although the lower bound is one, it is considered safer to use `LBound` in case it changes in the future.

Tip

The macro in Listing 5 is careful and checks to see if the argument is an array or a single argument. The macro does not verify that each value is numeric. You may be as careful as you like. The more things you check, the more robust the macro is, but the slower it runs.

Passing one argument is as easy as passing two: add another argument to the function definition (see Listing 6). When calling a function with two arguments, separate the arguments with a comma; for example, `=TestMax(3, -4)`.

Listing 6. TestMax accepts two arguments and returns the larger

```
Function TestMax(x, y)
  If x >= y Then
    TestMax = x
  Else
    TestMax = y
  End If
End Function
```

Arguments are passed as values

Arguments passed to a macro from Calc are always values. It is not possible to know what cells, if any, are used. For example, `=PositiveSum(A3)` passes the value of cell A3, and *PositiveSum* has no way of knowing that cell A3 was used. If you must know which cells are referenced rather than the values in the cells, pass the range as a string, parse the string, and obtain the values in the referenced cells.

Writing macros that act like built-in functions

Although Calc finds and calls macros as normal functions, they do not really behave as built-in functions. For example, macros do not appear in the function lists. It is possible to write functions that behave as regular functions by writing an Add-In. However, this is an advanced topic that is for experienced programmers and is beyond the scope of this guide. Some information, along with links to more detailed reading, is available in the Help.

Deleting LibreOffice Basic macros

Use the following steps to delete an unwanted macro:

1) Use **Tools > Macros > Organize Macros > Basic** on the Menu bar to open the Basic Macros dialog (see Figure 442 on page 428).
2) Select the macro to be deleted and click the **Delete** button.
3) Calc displays a confirmation dialog. Click **Yes** to continue.
4) Click the **Close** button to remove the Basic Macros dialog from the screen.